A Keyword Search Based Enhanced Spatially Inverted Index List For The Health Data Retrieval

Sunil Kumar Reddy .P[1]* and Dr.P.Govindarajulu[2]
1*Department of Computer Science, Sri Venkateswara University, Tirupati, India
2Department of Computer Science, Sri Venkateswara University, Tirupati, India

Abstract
Retrieval of medical information from the large datasets is one of the highly demanding and crucial task in recent days. So, various research works aimed to develop an Information Retrieval (IR) system for determining the health data. Still, it remains with the major limitations of inefficient data handling, reduced accuracy, and increased searching error. Thus, this research work aims to develop a keyword based IR system by implementing an Enhanced Spatially Inverted Index List (ESIIL). In this system, the medical dataset, namely, disease symptom knowledge database is taken for analysis, which is preprocessed at the initial stage by performing the stop words removal and stemming processes. After that, the Parts of Speech (POS) tagging is used to extract the keywords from the query. Then, the probability scores, namely, Term Frequency (TF) and Inverse Document Frequency (IDF) are computed based on the similarity attributes. Here, the leaf to root based order of search is implemented, which improves the retrieval efficiency of searching. Moreover, two types of information retrieval is performed in this work such as most exactly matching documents, and partially matching documents related to the given query. The major benefits of this mechanism are reduced searching error, increased efficiency, and large dataset handling. In experiments, the performance results of the existing and proposed mechanisms are evaluated by using various measures. Also, the superiority of the proposed ESIIL is proved by comparing it with the existing techniques.

Key words:
Information Retrieval, Query Searching, Similarity Index (SI), Enhanced Spatially Inverted Index List (ESIIL), Term Frequency (TF), Inverse Document Frequency (IDF), and Parts of Speech (POS) Tagging.

1. Introduction

INFORMATION Retrieval is the process of retrieving the most relevant information from the large dataset based on the given query [1, 2]. It is extensively used in most of the application areas such as general applications, domain specific applications, and software applications, among them the medical information retrieval is one of the demanding application area [3-5]. Moreover, it simplifies the process of searching by identifying the similarity between the documents with respect to the keyword obtained from the query. Keyword search is a kind of searching mechanism that handles both the labeled and unlabeled data [6]. In which, the representation of non-labeled data have a length and continuous history. The index list is maintained during information retrieval for searching the information in an efficient way [7, 8].

1.1 Problem Identification

In search analysis, selecting the search number is highly difficult and vital challenge, in which the index based approaches are used for evaluating the search validity issue [9]. Moreover, numerous research works have been carried out to reorder and utilize the Reorder Dissimilarity Index (RDIs) in searching [10]. The traditional Searching Index (SI) algorithm is a kind of pair-wise dissimilarity index, in which the vector norm is used to convert the vertical form of object into a dissimilarity index [11]. If the vertical form of data in unavailable, different flexible dissimilarity metrics are utilized to convert the pair-wise relational data [12]. Then, the visual assessment of search tendency is used to estimate the quantity of searches [13]. Moreover, the index based approaches focused on the intra-search compactness and inter-search segregation, in which the additional factors such as statistical and geometrical properties are utilized [14]. The problems that exists in the traditional Search Tendency Index (SII) [15] is not able to handle large datasets, reduced accuracy, increased searching error, and high time complexity [16, 17]. In this work, the symptom of the disease is considered as a keyword entered by the user. For instance, some of the symptoms and its corresponding disease are shown in Table 1. When a user enter a keyword (i.e. symptom), the NLP tool is used to match the keyword with the database. If it is not exactly match, the relevancy check is enabled to find out the similar match. Based on the matching outcomes, the results are retrieved and provided to the user. In the existing systems, the results are predicted with better accuracy and increased false alarm rate. But, in the proposed system, this problem is overwhelmed by matching the similarity with the use of spatially inverted index list.

Manuscript received February 5, 2018
Manuscript revised February 20, 2018

Table 1: Disease and its symptom

Disease	Symptom
Hypertensive Disease	Chest pain
	Shortness of breath
	Dizziness
	Asthenia
	Fall
Diabetes	Polyuria
	Polydypsia
	Shortness of breath
	Chest pain
	Asthenia
Mental depression and disorder	Feeling hopeless
	Mood depressed
	Homelessness
Coronart Heart Disease	Angina pectoris
	Chest pain
	Sweating increased
Pneumonia	Cough
	Fever
	Yellow sputum
	Night sweat
Failure heart Congestive	Orthopnea
	Rale
	Chill
	Green sputum

1.2 Objectives

The major research objectives of this paper are as follows:

To preprocess the given medical data, the stemming and stop words removal processes are performed.

To extract the keywords from the query, the Parts of Speech (POS) tagging is used, which separates the sentence into a noun, adverb, adjective, and etc.

To estimate the probability score, the Term Frequency (TF) and Inverse Document Frequency (IDF) are computed.

To form the index for retrieving the information, an Enhanced Spatially Inverted Index List (ESIIL) is formed, which retrieves the information in the leaf to root order.

1.3 Organization

The rest of the sections in the paper are structured as follows: the existing techniques and algorithms that used to perform the query searching based information retrieval are surveyed in Section II. The description about the proposed information retrieval mechanism with its clear flow and algorithm are presented in Section III. The experimental results of the existing and proposed information retrieval schemes are analyzed and compared by using various performance measures in Section IV. Finally, the paper is concluded and the future work that can be implemented in future are stated in Section V.

2. Related Works

In this section, the existing techniques and algorithms related to web service recommendation are surveyed with its advantages and disadvantages.

Pandya, et al [18] implemented a searching mechanism by integrating the indexing tools and techniques for multi-format data. The suggested architecture has three tiers, which includes data acquisition, middle tier or application tier. In which, the data acquisition layer was considered as a bottom layer that collects the processed data. Then, the middle layer acts like a heart of searching, which has the master index. Finally, the application layer was used to enable the search automation process, in which the user fires a query to the middle layer. The advantages that observed from this paper were increased efficiency and reduced time consumption. *Wu, et al* [19] developed a new authentication data structure for processing the spatial keyword queries. Here, the Verification Object (VO) was designed to authenticate the safe zones and top k-results in the query. Moreover, an enhanced data structure was implemented to reduce the communication cost. In this design, the client extracted the top k-result from VO for authentication, and the ranking score was computed for all the objects in VO. Here, the safe zone verification was mainly performed to check the missing objects. Moreover, a naïve approach was utilized to compute the correct zone for transferring the whole dataset, which reduced communication cost.

- *Yan, et al* [20] introduced an optimized document ordering mechanism for reducing the inverted index size and increasing the query throughput. The major contributions of this paper were as follows:
- The compressibility of frequency values were improved by using the move-to-front coding technique.
- The impact of docID was studied based on the index size and query throughput.
- Moreover, the trade-off between the speed and compression ratio was estimated by analyzing various compression techniques.
- Also, this paper optimized some other methods such as Gamma Diff, Interpolative Coding (IPC) and S16-128. However, it has the major limitations of increased complexity and reduced computation efficiency. *Velusamy*, et al [21] suggested a compressed inverted index data structure for mapping a file in a word or a set of documents. Here, the basic factors such as storage technique, index size, maintenance, fault tolerance, merge factors and look up speed were considered. The limitation behind this work was, it has an increased time consumption during the process of mapping. *Borse, et al* [22] suggested a fast nearest neighbor algorithm for managing the multi-dimensional spatial data based on an inverted index. Typically, the index data structure was

used to store the contents of the original in a compressed format. Here, the inverted index was developed to enable the fast searching of text with increased speed and reduced time consumption. Moreover, the operations such as union and intersection were performed to support the query semantics. The major benefits of this paper were reduced number of intervals and space cost in an inverted index. *Patil* [23] investigated different searching techniques for enabling an efficient query search with the use of spatial inverted index. The techniques that studied in this paper were IR2 – Tree, hybrid index structures, and spatial keyword queries. The demerits of these techniques were also examined in this paper for selecting the most suitable method. From the paper, it is analyzed that the searching nearest neighbor technique was the most suitable technique. *Lin, et al* [24] suggested a distributed visual retrieval system for generating a codebook with reduced memory and time consumption. Here, a vector space decomposition technique was utilized to improve the performance of product quantization. In this system, the time efficiency was improved by developing a distributed framework.

Gupta, et al [8] introduced a mixed script Information Retrieval (IR) process based on the query logs. In this system, a principled solution as provided to handle both the term matching and spelling variation in the document. *Gupta, et al* [25] used a fuzzy logic based ranking function for efficiently retrieving the information. The intention of this paper was to fetch the relevant documents from the large datasets based on the ranking function. Here, the term weighting schema was designed based on the measures of term frequency, inverse document frequency and normalization. *Kopliku, et al* [26] utilized an information retrieval paradigm by using the cross vertical aggregated search and relational aggregated search mechanisms. The authors identified five basic aggregation operations such as sorting, grouping, merging, splitting and extracting for result aggregation.

2.1 Visual Assessment of Searching Tendency

There are numerous probable methods are used to obtain the RDI, which uses the SI of the unlabeled data (i.e) secured inputs using SI-Index algorithm. Let consider, the number of objects $O = \{O1, O2 \dots On\}$ in the data (such as, groceries, fishes, flowers, beers, etc.). Then, the vertical data is denoted as $F = \{f1, f2 \dots fn\}\ fi \subset Rh$, in which each coordinate of the fi vector gives a feature value of every h attribute (aj, j=1, 2 ... h) that corresponding to the object Oi. Typically, the relational data is captured in a direct manner, i.e. pair-wise similarities / dissimilarities amongst objects, which denoted as $n \times n$ symmetric index D. The vertical data F is converted into inequalities or dissimilarities, which is illustrated as follows: $D = [dij = \|\ fi - fj\ \|, 1 \leq i, j \leq n]$. Generally, the dissimilarity Indexes satisfy $1 \geq dij \geq 0$; $dij = dji$; $dii = 0$, for $1 \leq i, j \leq n$. Moreover, the dissimilarity index D is displayed as a gray scale data by using the SI algorithm. In which, every element is a scaled based on the dissimilarity value dij between objects Oi and Oj. In data smoothening, the sum of dark blocks in any RDI and the sum of major keywords are equal. So, various sophisticated algorithms are used for an effective information retrieval.

From the survey, it is examined that the existing techniques have both advantages and disadvantages, but it mainly lacks with the following limitations:

- Increased computational complexity
- High searching time
- Inefficient processing
- Not highly suitable for handling a larger datasets

To solve these problems, this paper aims to develop a new query based information retrieval system by developing an inverted index list mechanism.

3. Proposed Methodology

In this section, the description about the proposed methodology is presented with clear flow illustration. The motive of this paper is to retrieve the information based on the query from the medical dataset. Here, an inverted index matching mechanism is implemented for an efficient query processing. The working flow of the proposed system is shown in Fig 1 and working procedure is illustrated in Algorithm I, which includes the following stages:

- Preprocessing
- Keyword extraction
- Index list generation
- Ranking
- Score estimation
- Information retrieval

At first, the medical dataset obtained from the disease symptom knowledge database [27] is taken as the input, which is preprocessed by performing the stemming and stop words removal processes. Then, the keywords are extracted from the query by using the POS tagging, and the probability score is computed based on the Term Frequency (TF) and Inverse Document Frequency (IDF) measures. Based on this, the index list is formed and the data inside the index are ranked. Consequently, the similar indexed words are generated with respect to the upper and lower bound index. The score is also calculated with respect to the query and document data, and finally the similar indexed data that exactly or partially matched to the query is retrieved.

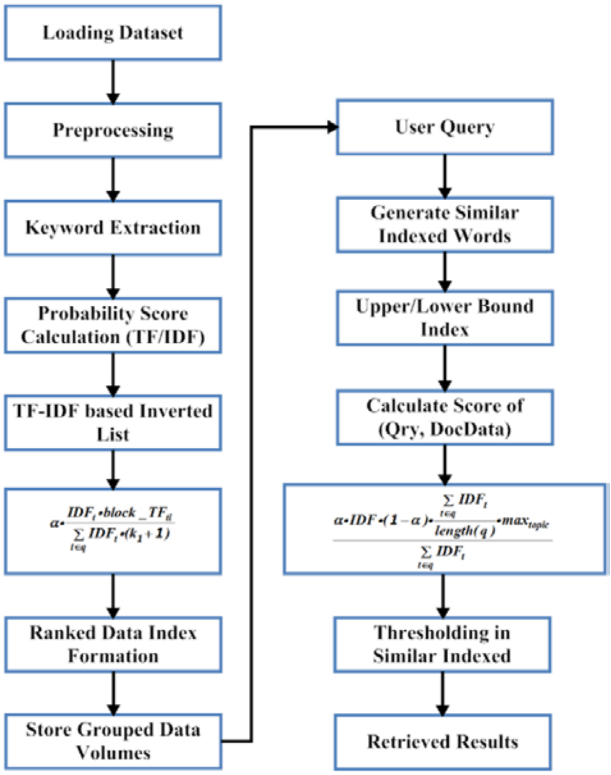
Loading Dataset
Preprocessing
Keyword Extraction
Probability Score Calculation (TF/IDF)
TF-IDF based Inverted List
$\alpha \cdot \frac{IDF_t \cdot block\_TF_{tl}}{\sum_{t \in q} IDF_t \cdot (k_1 + 1)}$
Ranked Data Index Formation
Store Grouped Data Volumes
User Query
Generate Similar Indexed Words
Upper/Lower Bound Index
Calculate Score of (Qry, DocData)
$\frac{\alpha \cdot IDF \cdot (1-\alpha) \cdot \frac{\sum_{t \in q} IDF_t}{length(q)} \cdot max_{topic}}{\sum_{t \in q} IDF_t}$
Thresholding in Similar Indexed
Retrieved Results

Fig. 1 Flow of the proposed system

Algorithm I – Knowledge Database Creation

```
Step 1:  var acc = 0 //All medical based document in the list is stored;
Step 2:  Load documents from the selected directory with the sub-directories;
Step 3:  For each root/directory/files in the absolute path
Step 4:      $MAXc = Max(list[i].CurrentDoc) \mid\mid \forall i\ 0 \le i \le DocList.size()$
Step 5:      while $i < size(DocList)$ do
Step 6:          if $DocList.item(i)$ is file
Step 7:              $acc += Max(Retrived\ Doc\ List)$
Step 8:              $\forall j\ DocList[i].CurDoc \le j \le MAXc$
Step 9:          else // $DocList.item(i)$is a directory
Step 10:             Go to Step 5 with current directory;
Step 11:         end if;
Step 12: End for;
Step 13: For each doc in acc
Step 14:     Split words and apply the stemming process;
Step 15:     Apply the stop words removal process
Step 16:     $Max\left(\frac{IDF_{t_i} \cdot block\_TF_{t_{i,j}}}{\sum_{t_i \epsilon q} IDF_{t_i} \cdot (k_1+1)} \forall j\ list[i].curDoc \le j \le cur\ Max\right)$
Step 17:     Calculate IDF;
Step 18:     Index free construction;
Step 19:     Update the index tree for the current document/word score;
Step 20:     Return index list;
```

3.1 Preprocessing

In this stage, the medical dataset, namely, symptom knowledge database is loaded, which contains the attributes such as disease name and its symptom. Then, it is preprocessed by performing the processes such as stemming and stop words removal. In which, stemming is defined as the process of mapping the morphological variations of words into a common word. The stemming is widely used in many information retrieval applications for improving the performance of retrieval efficiency. The main reason of using stemming is to reduce the size of index file, in which the dictionary look-up is used to perform the stripping process. During this process, the stemmer aims to obtain the stem of a word that carry the lexical information about the word. After stemming, the stop words are removed, in which the words include wh words, conjunction words, auxiliary verbs, articles, prepositions, pro-nouns and etc. These words must be removed from the dataset for reducing the searching time of query processing. In most of the information retrieval system, the dimensionality of term space can be reduced by removing these words. Then, the knowledge database is constructed by adding the table with the attributes of disease name and preprocessed words.

3.2 Keywords Extraction

After preprocessing, the POS tagging is applied to extract the keywords from the preprocessed data, in which each word is assigned in a sentence. Typically, the POS contains some important grammatical information that selects a most probable speech sequence of the words in the sentence. Also, it is used to separate the sentence into a noun, adverb, verb, adjective, and all parts of speech tag. In this stage, the user registration is initiated by maintaining the history of queries corresponding the user. Then, the query is parsed from the user and split into multiple words, and the stop words are again removed from the query by using the Stanford-postagger NLP tool. Consequently, the spatial similar words are generated by using the wordnet dictionary tool.

3.3 Term Frequency and Inverse Document Frequency Computation

In this stage, the random user entry is generated with respect to the random keyword queries, which is corresponding to the user history. After that, both the index and inverted index tree are created for each document based on the unique keywords score, which is calculated based on the TF and IDF values. Then, the grouped volume data is stored into the knowledge database. The ranking function contains both the TF and IDF score values, which are evaluated for demonstrating the better retrieval performance.

$$\mathrm{Score}\ (\mathrm{q},\mathrm{d}) = \ \alpha \cdot \mathrm{Score}_{\mathrm{BM25}}(\mathrm{q},\mathrm{d}) + (1-\alpha) \cdot \mathrm{Score}_{\mathrm{topic}}(\mathrm{q},\mathrm{d}) \qquad (1)$$

Where, score (q, d) indicates the relevance score of the document *d* that corresponding to the query q, α represents the trading factor, $Score_{BM25}(q,d)$ represents the score of TF and IDF, and $Score_{topic}(q,d)$ denotes the score of topic. In this paper, the TF and IDF values are estimated by using the term matching mechanism that used for information retrieval. Consequently, the score of BM25 with respect to the document *d* and term *t* is computed as follows:

$$\mathrm{Score}_{\mathrm{BM25}}(\mathrm{t},\mathrm{d}) = \mathrm{IDF}_{\mathrm{t}} \cdot \mathrm{TF}_{\mathrm{BM25}}(\mathrm{t},\mathrm{d}) \qquad (2)$$

Then, the measures IDF_t and $TF_{BM25}(t,d)$ are computed as follows:

$$\mathrm{IDF}_{\mathrm{t}} = \log\left(\frac{\mathrm{N}}{\mathrm{N}_{\mathrm{t}}}\right) \qquad (3)$$

Where, *N* indicates the total number of documents, and N_{t} denotes the number of documents that contains the term *t*.

$$\mathrm{TF}_{\mathrm{BM25}}(\mathrm{t},\mathrm{d}) = \frac{\mathrm{f}_{\mathrm{t,d}} \times (\mathrm{k}_1+1)}{\mathrm{f}_{\mathrm{t,d}} + \mathrm{k} \times ((1-\mathrm{b}) + \mathrm{b} \times (\mathrm{l}_{\mathrm{d}}/\mathrm{l}_{\mathrm{avg}}))} \qquad (4)$$

Where, $f_{t,d}$ indicates the frequency term, l_d represents the length of document, l_{avg} represents the average length of documents, *k* and *b* are the default values as 1.2 and 0.75. The scores

$$\mathrm{Score}_{\mathrm{BM25}}(\mathrm{q},\mathrm{d}) = \left(\frac{\sum_{\mathrm{t}\in\mathrm{q}} \mathrm{IDF}_{\mathrm{t}} \cdot \mathrm{TF}_{\mathrm{BM25}}(\mathrm{t},\mathrm{d})}{\sum_{\mathrm{t}\in\mathrm{q}} \mathrm{IDF}_{\mathrm{t}} \cdot (\mathrm{k}_1+1)}\right) \qquad (5)$$

$$\mathrm{Score}_{\mathrm{topic}}(\mathrm{t},\mathrm{d}) = \mathrm{P}(\mathrm{d}|\mathrm{t}) \propto \mathrm{P}(\mathrm{t}|\mathrm{d})\mathrm{P}(\mathrm{d}) \qquad (6)$$

$$\text{Where, } \mathrm{P}(\mathrm{d}|\mathrm{t}) = \sum_{\mathrm{k}=1}^{\mathrm{K}} \emptyset_{\mathrm{kt}} \cdot \theta_{\mathrm{dk}} \qquad (7)$$

In this algorithm, it is assumed that the prior of document is equal for all the documents, so the normalized topic score is estimated as follows:

$$\mathrm{Score}_{\mathrm{topic}}(\mathrm{q},\mathrm{d}) = \frac{\sum_{\mathrm{t}\in\mathrm{q}} \sum_{\mathrm{k}=1}^{\mathrm{K}} \emptyset_{\mathrm{kt}} \cdot \theta_{\mathrm{dk}}}{\mathrm{length}\ (\mathrm{q})} \qquad (8)$$

Where, the length (q) indicates the number of terms in the query. At last, the TF and IDF values are integrated to calculate the relevance score of the document, which is shown in below:

$$\mathrm{Score}_{(\mathrm{q},\mathrm{d})} = \alpha \cdot \frac{\sum_{\mathrm{t}\in\mathrm{q}} \mathrm{IDF}_{\mathrm{t}} \cdot \mathrm{TF}_{\mathrm{BM25}}(\mathrm{t},\mathrm{d})}{\sum_{\mathrm{t}\epsilon\mathrm{q}} \mathrm{IDF}_{\mathrm{t}} \cdot (\mathrm{k}_{\mathrm{t}}+1)} + (1-\alpha) \cdot \sum_{\mathrm{t}\in\mathrm{q}} \frac{\mathrm{Score}_{\mathrm{topic}}(\mathrm{t},\mathrm{d})}{\mathrm{length}\ (\mathrm{q})} \qquad (9)$$

Based on the values of TF and IDF, the spatially inverted index list is formed for an efficient information retrieval.

3.4 Inverted Index List Formation

In this step, the category of the given query is identified in order to retrieve the first level of matched documents. If the queries are not exactly match with the category, the documents that are partially matched with the query are retrieved. In the proposed system, the leaf to root based order of search is performed for information retrieval. Also, the inverted index list simplifies the process of searching by exactly matching the query with the attribute.

Table 1: Notation description

SC_{w}	Similarity Score value for keyword w
q	Q number of attributes in given query
kw	kw number of key words in database
$\mathrm{SIM}(\mathrm{A}_{\mathrm{i}}, \mathrm{A}_{\mathrm{j}})$	Similarity between the ith attribute in query and jth keyword.
n	N number of synonyms of attribute
m	M number of synonyms for query
$\mathrm{A}_{\mathrm{i}}^{\mathrm{x}}$	ith query attribute xth similar word
$\mathrm{A}_{\mathrm{i}}^{\mathrm{y}}$	j th keyword attribute yth similar word
$\mathrm{VF}(\mathrm{S}_{\mathrm{a}}, \mathrm{A}_{\mathrm{i}})$	Visiting frequency for ath keyword for the ith query attribute
$\mathrm{L}_{\mathrm{p.}}$	P number of Document class’s
c	Count frequency
$\mathrm{VF}(\mathrm{S}_{\mathrm{a}})$	Visiting frequency for ath service for all learners

Here, the similarity between the query attribute and the keyword attribute are estimated for finding the similar word, which is shown in below:

$$\mathrm{SIM}(\mathrm{A}_{\mathrm{i}}, \mathrm{A}_{\mathrm{j}}) = \frac{\sum_{\mathrm{x}=1}^{\mathrm{n}} \sum_{\mathrm{y}=1}^{\mathrm{m}} \mathrm{f}(\mathrm{A}_{\mathrm{i}}^{\mathrm{x}}, \mathrm{A}_{\mathrm{j}}^{\mathrm{y}})}{\mathrm{n}} \qquad (1)$$

$$f(A_i^x, A_j^x) = \begin{cases} 1 & if\ A_i^x == A_j^x \\ 0 & else \end{cases} \qquad (2)$$

If both attributes are same, the value is assigned as 1; otherwise, it is assigned as 0; Consequently, the visiting frequency is estimated based on the number of document classes, and similar word, which is depicted as follows:

$$VF(S_i) = \frac{\sum_{x=0}^{p} f(L_p, S_i)}{p} \qquad (3)$$

Then, the visiting frequency for ath keyword for the ith query attribute is computed with respect to P number of document classes as shown in below:

$$\mathrm{VF}(\mathrm{S}_{\mathrm{i}}, \mathrm{A}_{\mathrm{i}}) = \sum_{\mathrm{x}=0}^{\mathrm{p}} \mathrm{f}(\mathrm{L}_{\mathrm{p},}\mathrm{S}_{\mathrm{i}}, \mathrm{A}_{\mathrm{i}}) \qquad (4)$$

From that, the counting frequency between the values of $(L_{p,}S_i)$ and (L_p, S_i, A_i) is computed, if the value of c is greater than the value of 0, the actual counting frequency is assigned; otherwise, it is assigned as 0.

$$f(L_{p,}S_i) = \begin{cases} c & \text{if } c > 0 \\ 0 & \text{else} \end{cases} \quad (5)$$

$$f(L_{p,}S_i, A_i) = \begin{cases} c & \text{if } c > 0 \\ 0 & \text{else} \end{cases} \quad (6)$$

Then, the rank is computed with respect to the similarity attribute, and the visiting frequency, which is shown in below:

$$R(S_i) = \sum_{x=0}^{p} r_i^x \quad (7)$$

$$SC_w = \sum_{i=0}^{q} \sum_{j=0}^{s} SIM(A_i, A_j) + \sum_{i=0}^{q} VF(S_w, A_i) + VF(S_w) + R(S_w) \quad (8)$$

Finally, the exactly matched documents are retrieved at the first level; if it is found, the second order documents that is partially matched with the query is retrieved.

Algorithm II – Inverted Indexing

```
Step 1: Initialize the measures as Accum = 0, and gap_accum = 0;
Step 2: Consider that the Cur Max is the last document in the heap;
Step 3: The highest value is selected by using the function of Max;
Step 4: curMax = $Max\ (list\ [j].curDoc | \forall j 0 \le j < list.size)$
        i = 0;
Step 5: Process all the inverted lists;
Step 6: while $i < list.size$ do
Step 7: if list [i] is the TF-IDF based inverted list then
            $accum += \alpha. Max\left(\frac{IDF_{ti} \cdot block\_TF_{ti,j}}{\sum_{ti \in q} IDF_{ti} \cdot (k_1+1)} | \forall j\ list[i].cuDoc \le j \le curMax\right)$
        Else
            $accum += (1-\alpha). Max\left(\frac{block\_Topic_{ti,j}}{length(q)} | \forall j\ list[i].cuDoc \le j \le curMax\right)$
        $gap\_accum += (1-\alpha). Max\left(\frac{block\_Topic_{ti,j}}{length(q)} | \forall j\ list[i].cuDoc \le j \le curMax\right)$
Step 8: if $(accum > threshold\ and\ gap\_accum > gap)$
            Or $i == (list.size - 1)$
            Break;
        $i++$;
Step 9: Return list [i];
```

The major benefits of the proposed ESIIL are as follows:

It has the ability to process the lager datasets

It does not depend on the representation of data for searching

Filtering and data smoothening enhanced the performance of searching efficiency

4. Performance Analysis

This section presents the performance evaluation results of both existing and proposed techniques with respect to various performance measures. In this simulation, the HBase region server is executed on the same machines as DataNodes and ZooKeeper with the ensemble of 3 machines. Then, the spatial query processing time is estimated for each query and its settings. Moreover, the disease symptom knowledge database is utilized to evaluate the performance of this system. The measures used to test the results are execution time, precision, recall, f-measure, query processing time, and correctness. The simulation setting of this environment in shown in Table 2.

Table 2: Simulation settings

Parameter	***Definition***
Hadoop version	hadoop-2.6.5
Running tool	jdk1.8.0_161
Number of nodes	10
Master machine	1
RAM	12GB
Processor	2.4 GHz 64-bits quad core Xeon E5530
Disks	Two 7200 rpm SATA
Disk size	500GB and 250 GB

4.1 Execution Time

Execution time is defined as the amount of time required to execute the given process. Here, the execution time of the information retrieval process is calculated with respect to the data size in terms of GB, which is shown in Fig 2. The existing techniques considered for this analysis are Inverted Index (II), and Spatial Inverted Index (SII). It is calculated as follows:

$$\text{Execution Time} = \text{Ending time of the process} - \text{Starting time of the process} \quad (5)$$

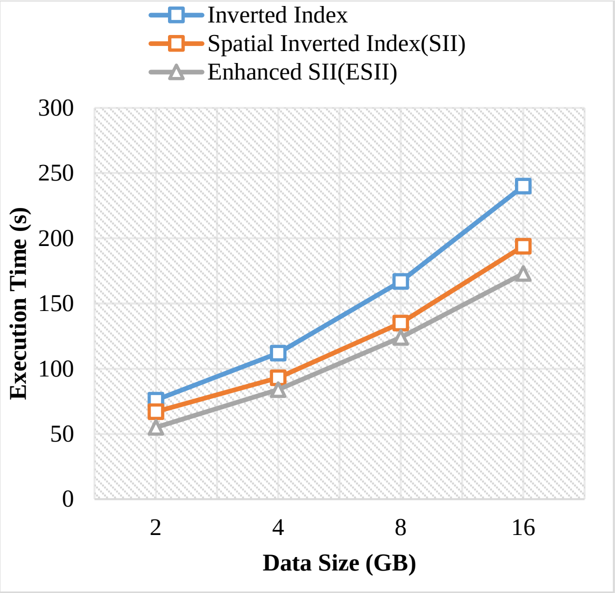

Fig. 2 Execution time

When compared to existing techniques, the proposed ESIIL requires the reduced execution time. Because, the searching time of IR is reduced by finding the TF and IDF scores, which simplifies the process of searching.

4.2 Precision, Recall and F-Measure

Precision and recall are also used to evaluate the effectiveness of information retrieval. Precision is defined as the positive predictive value that provides the results relevant to a retrieval process. Then, it is estimated based on the ratio of true positives and true positives plus false positives. Recall is estimates the effectiveness of IR by computing the most relevant results. Based on the values of TP, TN, FP and FN, the precision and recall are estimated as shown in the following equations:

$$\text{Precision} = \frac{\text{TP}}{(\text{TP+FP})} \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP+FN})} \tag{5}$$

The F-measure is defined as the process of calculating the test's accuracy and it is otherwise stated as the weighted arithmetic mean of precision and recall. Fig 3 shows the precision, recall and f-measure analysis of the existing II, SII and proposed ESIIL techniques. Also, the % of precision improvement is estimated with respect to the number of documents as shown in Fig 4. From this, it is observed that the proposed technique provides the better precision, recall, and f-measure values, when compared to the other techniques.

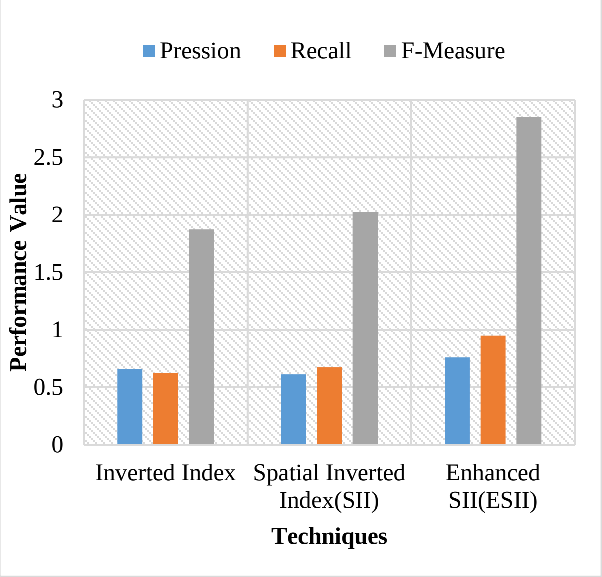

Fig. 3 Analysis of precision, recall and f-measure

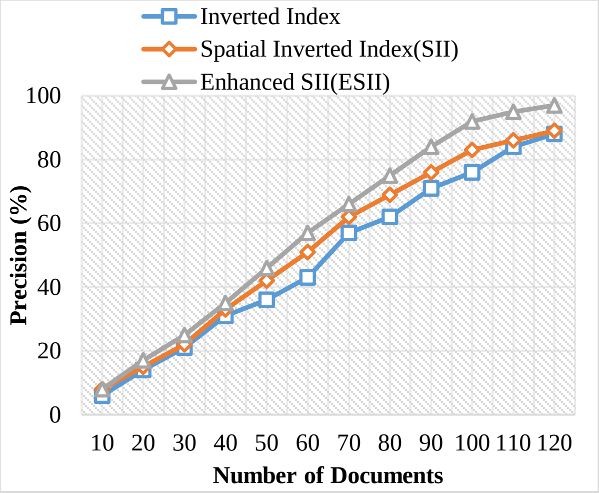

Fig 4. % of improvement in precision

4.3 Average Query Processing Time

The average query processing time is defined as the amount of time difference from the query receiving time and the finishing time of an information retrieval. Here, the query processing time is calculated for both existing and proposed techniques with respect to varying query length. It is calculated as follows:

$$Average\ query\ processing\ time = Query\ receiving\ time - Information\ Retrieval\ Time \tag{1}$$

Fig 5 evaluates the average query processing time of both existing and proposed system with respecting the query length ranging from 3 to 10, based on this value, the time in terms of seconds is calculated. When compared to the existing II and SII techniques, the proposed ESIIL provides the reduced time consumption.

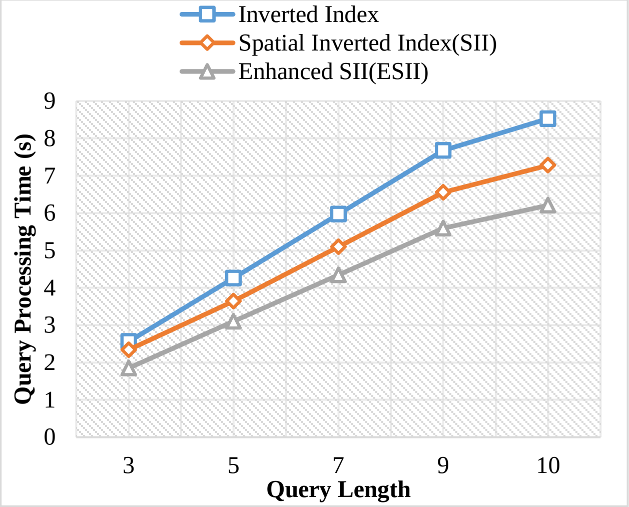

Fig. 5 Average query processing time

4.4 Correctness

The percentage of correctness of the existing and proposed techniques are evaluated in this analysis with respect to varying query length as shown in Fig 6. Here, the % of correctly answered questions are computed for proving the efficiency of the proposed ESIIL technique. When compared to the existing techniques, the count of proposed ESIIL is linearly increased with the increase in the query length. Because, the stemming and stop words removal processes are performed in the proposed system, which reduces the length of query. Also, the POS tagging can extract the exact keyword for matching the documents, which reduces the complexity of searching. Thus, the correctness of the proposed system is better than the other techniques.

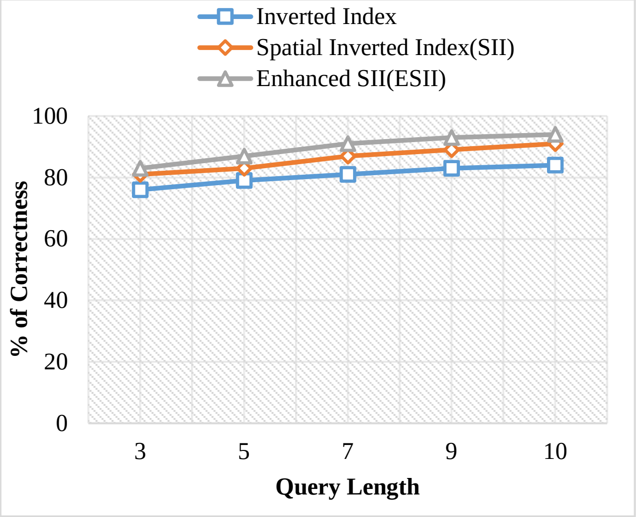

Fig. 6 % of Correctness

5. Conclusion and Future Work

In this paper, a new IR system, namely, ESIIL is proposed for processing the medical dataset. It mainly aims to reduce the complexity of searching by matching the keyword with the similarity attribute. Here, the disease symptom knowledge database is used, which is preprocessed by applying the stemming and stop words removal processes. Then, the keywords from the filtered data are extracted by using the POS tagging, which reduces the complexity of searching. Moreover, the TF and IDF scores are computed to rank the documents that exactly match with the query. Based on these score values, the inverted index list is constructed, in which the leaf to root of search is performed for retrieving the corresponding information. The proposed ESIIL is a kind of parameter free methodology for an automated estimation of number of searches within the non-labeled datasets. Also, it explores the distance metrics for improving the visualization of searching. The keywords and searching time are computed based on the projection of the word. The major advantages of this work are increased searching accuracy, reduced time consumption, and improved efficiency. During experimentation, the results of existing and proposed mechanisms are evaluated with respect to various measures such as execution time, precision, recall, f-measure, query processing time, and correctness. From the analysis, it is evident that the proposed ESIIL provides the better performance, when compared to the other techniques.

References

[1] N. Kanhabua, et al., "Temporal information retrieval," Foundations and Trends® in Information Retrieval, vol. 9, pp. 91-208, 2015.

[2] E. Chauhan and A. Asthana, "Review of Indexing Techniques in Information Retrieval," International Journal of Engineering Science, vol. 13940, 2017.

[3] K. Uthayan and G. Anandha Mala, "Hybrid Ontology for Semantic Information Retrieval Model Using Keyword Matching Indexing System," The Scientific World Journal, vol. 2015, 2015.

[4] D. Wolfram, "The symbiotic relationship between information retrieval and informetrics," Scientometrics, vol. 102, pp. 2201-2214, 2015.

[5] L. Weng, et al., "A privacy-preserving framework for large-scale content-based information retrieval," IEEE Transactions on Information Forensics and Security, vol. 10, pp. 152-167, 2015.

[6] K. Golub, et al., "A framework for evaluating automatic indexing or classification in the context of retrieval," Journal of the Association for Information Science and Technology, vol. 67, pp. 3-16, 2016.

[7] B. Saini, et al., "Information retrieval models and searching methodologies: Survey," Information Retrieval, vol. 1, p. 20, 2014.

[8] P. Gupta, et al., "Query expansion for mixed-script information retrieval," in Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, 2014, pp. 677-686.

[9] B. Koopman, et al., "Information retrieval as semantic inference: a Graph Inference model applied to medical search," Information Retrieval Journal, vol. 19, pp. 6-37, 2016.

[10] G. Singh and V. Jain, "Information retrieval (IR) through semantic web (SW): An overview," arXiv preprint arXiv:1403.7162, 2014.

[11] C. Lioma, et al., "Non-compositional term dependence for information retrieval," in Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 595-604.

[12] L. Derczynski, et al., "Time and information retrieval: Introduction to the special issue," Information Processing and Management, vol. 51, pp. 786-790, 2015.

[13] H. Hyman, et al., "A process model for information retrieval context learning and knowledge discovery," Artificial Intelligence and Law, vol. 23, pp. 103-132, 2015.

[14] M. Donge and V. Nandedkar, "Information Retrieval using Context Based Document Indexing," International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), vol. 2, 2014.

[15] S. P. Reddy and P. Govindarajulu, "Implementation of Nearest Neighbor Retrieval," International Journal of Computer Sciences and Engineering, vol. 5, pp. 51-57, 2017.

[16] G. H. Yang, et al., "Dynamic information retrieval modeling," Synthesis Lectures on Information Concepts, Retrieval, and Services, vol. 8, pp. 1-144, 2016.

[17] P. Sunil Kumar Reddy and P. Govindarajulu, "Adjacent Search Outcomes with Keywords," International Journal of Computer Science and Information Technologies, vol. 7, pp. 312-317, 2016.

[18] H. D. Pandya, et al., "SearchAutomaton: Searching mechanism for multi-format data by combining indexing tools and techniques," in Proceedings of the International Conference on Advances in Information Communication Technology & Computing, 2016, p. 67.

[19] D. Wu, et al., "Authentication of moving top-k spatial keyword queries," IEEE Transactions on Knowledge and Data Engineering, vol. 27, pp. 922-935, 2015.

[20] H. Yan, et al., "Inverted index compression and query processing with optimized document ordering," in Proceedings of the 18th international conference on World wide web, 2009, pp. 401-410.

[21] K. Velusamy, et al., "Inverted indexing in big data using hadoop multiple node cluster," International Journal of Advanced Computer Science and Applications, vol. 4, pp. 156-161, 2013.

[22] S. Borse and P. Chawan, "Inverted Index for Fast Nearest Neighbour," International Journal of Computer Science and Mobile Computing, vol. 5, pp. 331-336, 2016.

[23] S. B. Patil, "Survey of Searching Nearest Neighbor Based on Keywords using Spatial Inverted Index," International Journal of Engineering Research and General Science, vol. 2, 2014.

[24] X. Lin, et al., "The distributed system for inverted multi-index visual retrieval," Neurocomputing, vol. 215, pp. 241-249, 2016/11/26/ 2016.

[25] Y. Gupta, et al., "A new fuzzy logic based ranking function for efficient Information Retrieval system," Expert Systems with Applications, vol. 42, pp. 1223-1234, 2015/02/15/ 2015.

[26] A. Kopliku, et al., "Aggregated search: A new information retrieval paradigm," ACM Computing Surveys (CSUR), vol. 46, p. 41, 2014.

[27] (2004). Disease-Symptom Knowledge Database. Available: http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html

P.Sunil Kumar Reddy received MCA from Bharathiar University in 2004 and M.Phil. Computer Science from Madurai Kamaraj University. He is Pursuing Ph.D. in SV University Tirupati.His research area are Databases and Data Mining.

Dr.P.Govindarajulu Professor, Department of Computer Science, Sri Venkateswara University, Tirupati, India. He received his M.Tech.from IIT Bombay (Mumbai). His area of research: Databases, Data Mining, Image Processing, Intelligent Systems and Software Engineering, Parallel Computing.